|  |  |
| --- | --- |
| 读文件 | 计算 |
| sem empty = 2, full=0, lock=1  read\_from\_file  num = read\_file()  wait(empty)  wait(lock)  queue.push(num)  signal(lock)  signal(full) | **compute**  **for i =1 to 5**  **for j =1 to 2**  **wait(full)**  **wait(lock)**  **value[j]=queue.pop()**  **signal(lock)**  **signal(empty)** |

**#include** "stdio.h"

**#include** <pthread.h>

**#include** <semaphore.h>

**#include** <stdlib.h>

**#include** <time.h>

**#define** \_\_compute() ((**rand**() % 2) **>=** 1 **?** 0 **:** 1)

**void** **compute**(int a, int b)

{

**if** (**\_\_compute**())

    {

**printf**("%d \* %d = %d\n", a, b, a \* b);

    }

**else**

    {

**printf**("%d + %d = %d\n", a, b, a + b);

    }

}

**#define** datatype **int**

typedef struct

{

    datatype \*arr;

**int** front, tail;

**int** cap, len;

} Queue;

Queue **NewQueue**(int cap)

{

    Queue q = {

        .front = 0,

        .tail = 0,

        .len = 0,

        .cap = cap,

        .arr = (datatype \*)**malloc**(sizeof(datatype) \* cap),

    };

**return** q;

}

**int** **queue\_full**(Queue \*q)

{

**return** (q**->**cap **==** q**->**len);

}

**int** **queue\_empty**(Queue \*q)

{

**return** (q**->**len **==** 0);

}

**void** **queue\_push\_back**(Queue \*q, datatype element)

{

**if** (**queue\_full**(q))

    {

**return**;

    }

    q**->**arr[q**->**tail] = element;

    q**->**tail = (q**->**tail + 1) % q**->**cap;

    q**->**len++;

}

datatype **queue\_pop\_head**(Queue \*q)

{

**if** (**!queue\_empty**(q))

    {

        datatype tmp = q**->**arr[q**->**front];

        q**->**front = (q**->**front + 1) % q**->**cap;

        q**->**len--;

**return** tmp;

    }

}

Queue \*queue;

**int** value[2] = {0, 0};

sem\_t empty, full, mutex;

pthread\_t r1, r2, c1, c2;

**void** \***read\_file**(void \*file\_name)

{

**char** \*filename = (**char** \*)file\_name;

    FILE \*r\_file = **fopen**(filename, "r");

**if** (r\_file **==** **NULL**)

    {

**return** **NULL**;

    }

**int** num = 0;

**char** tmp[1024];

**for** (**int** i = 0; i **<** 10; i++)

    {

**fgets**(tmp, 1024, r\_file);

        num = **atoi**(tmp);

**sem\_wait**(&empty);

**sem\_wait**(&mutex);

**queue\_push\_back**(queue, num);

**sem\_post**(&mutex);

**sem\_post**(&full);

    }

}

**void** \***consumer**(void \*args)

{

**int** i, j;

**for** (i = 0; i **<** 5; i++)

    {

**for** (j = 0; j **<** 2; j++)

        {

**sem\_wait**(&full);

**sem\_wait**(&mutex);

            value[j] = **queue\_pop\_head**(queue);

**sem\_post**(&mutex);

**sem\_post**(&empty);

        }

**int** a = value[0];

**int** b = value[1];

**compute**(a, b);

    }

}

**int** **main**()

{

**srand**((**int**)**time**(0));

**int** t1, t2, t3, t4;

    Queue queue1 = **NewQueue**(2);

    queue = &queue1;

**sem\_init**(&empty, 0, 2);

**sem\_init**(&mutex, 0, 1);

**sem\_init**(&full, 0, 0);

    t1 = **pthread\_create**(&c1, **NULL**, consumer, **NULL**);

    t2 = **pthread\_create**(&c2, **NULL**, consumer, **NULL**);

    t3 = **pthread\_create**(&r1, **NULL**, read\_file, (**void** \*)"1.dat");

    t4 = **pthread\_create**(&r2, **NULL**, read\_file, (**void** \*)"2.dat");

**pthread\_join**(r1, **NULL**);

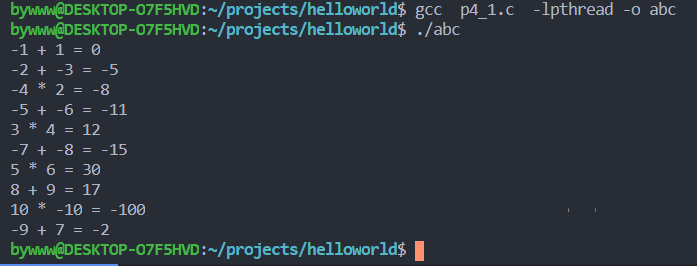
**pthread\_join**(r2, **NULL**);

**pthread\_join**(c1, **NULL**);

**pthread\_join**(c2, **NULL**);

**return** 0;

}



**#include** "stdio.h"

**#include** <pthread.h>

**#include** <semaphore.h>

**#include** <stdlib.h>

**int** flag = 0;

**void** **compute**(int a, int b)

{

**if** (flag **==** 1)

    {

**printf**("%d \* %d = %d\n", a, b, a \* b);

    }

**else** **if** (flag **==** 0)

    {

**printf**("%d + %d = %d\n", a, b, a + b);

    }

}

**int** value[2] = {0, 0};

sem\_t empty, full, mutex, empty2, full2;

pthread\_t r1, r2, c1, c2;

**void** \***read\_file**(void \*file\_name)

{

**char** \*filename = (**char** \*)file\_name;

    FILE \*r\_file = **fopen**(filename, "r");

**if** (r\_file **==** **NULL**)

    {

**return** **NULL**;

    }

**int** num = 0;

**char** tmp[1024];

**for** (**int** i = 0; i **<** 10; i++)

    {

**fgets**(tmp, 1024, r\_file);

        num = **atoi**(tmp);

**if** (filename **==** "1.dat")

        {

**sem\_wait**(&empty);

**sem\_wait**(&mutex);

            value[0] = num;

**sem\_post**(&mutex);

**sem\_post**(&full);

        }

**else** **if** (filename **==** "2.dat")

        {

**sem\_wait**(&empty2);

**sem\_wait**(&mutex);

            value[1] = num;

**sem\_post**(&mutex);

**sem\_post**(&full2);

        }

    }

}

**void** \***consumer**(void \*args)

{

**int** i, j, a, b;

**for** (i = 0; i **<** 5; i++)

    {

**for** (j = 0; j **<** 2; j++)

        {

**sem\_wait**(&full);

**sem\_wait**(&full2);

**sem\_wait**(&mutex);

            a = value[0];

            b = value[1];

**compute**(a, b);

            flag = 1^flag;

**sem\_post**(&mutex);

**sem\_post**(&empty);

**sem\_post**(&empty2);

        }

    }

}

**int** **main**()

{

**int** t1, t2, t3, t4;

**sem\_init**(&empty, 0, 1);

**sem\_init**(&empty2, 0, 1);

**sem\_init**(&mutex, 0, 1);

**sem\_init**(&full, 0, 0);

**sem\_init**(&full2, 0, 0);

    t1 = **pthread\_create**(&c1, **NULL**, consumer, **NULL**);

    t2 = **pthread\_create**(&c2, **NULL**, consumer, **NULL**);

    t3 = **pthread\_create**(&r1, **NULL**, read\_file, (**void** \*)"1.dat");

    t4 = **pthread\_create**(&r2, **NULL**, read\_file, (**void** \*)"2.dat");

**pthread\_join**(r1, **NULL**);

**pthread\_join**(r2, **NULL**);

**pthread\_join**(c1, **NULL**);

**pthread\_join**(c2, **NULL**);

**return** 0;

}

